Optimal Controller Design for the Inverted Double-Pendulum

Vedad Bassari

December 17, 2023

1 Introduction

The inverted pendulum is a common case study in control system design due to its unstable dynamics. This project analyzes an extension of this system, the dual inverted pendulum (Figure 1). The system of interest consists of a cart with mass $M = 1$ kg that moves along a single axis under the action of the input force $f(t)$. Two bars of masses $m_1 = m_2 = 0.2$ kg and lengths $l_1 = 0.5$ m, $l_2 = 1$ m are attached to the cart via hinges.

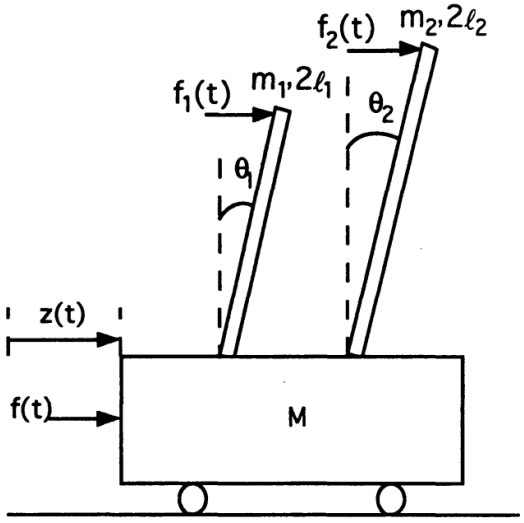


Figure 1: Free-body diagram of the dual pendulum system, adopted from [1].

The goal of the controller design is to stabilize these bars around their unstable vertical equilibrium. Desirable qualitative properties of the controlled system include a short response time ($t < 20$s), disturbance rejection with respect to low-frequency disturbances ($f1, f2 < 50$ N), and small control inputs ($f < 500$ N). Note that these design constraints are not specified by the problem statement, but imposed to roughly guide our design process. The following sections provide an analysis of the system's controllability and an optimal controller design that uses full-state feedback. Finally, we provide a theoretical overview of the system's closed-loop properties and design an observer-based optimal controller that accommodates realistic measurement constraints. All of the subsequent analysis is based on the linearized plant model presented by [1]:

$$\delta\dot{x}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-3M_1 g}{Z} & 0 & \frac{-3M_2 g}{Z} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{3g(Z+3M_1)}{4Zl_1} & 0 & \frac{9M_2 g}{4Zl_1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{9M_1 g}{4Zl_2} & 0 & \frac{3g(Z+3M_2)}{4Zl_2} & 0 \end{bmatrix} \delta x(t) + \begin{bmatrix} 0 \\ \frac{4}{Z} \\ 0 \\ \frac{-3}{Zl_1} \\ 0 \\ \frac{-3}{Zl_2} \end{bmatrix} \delta u(t)$$

$$\delta y(t) = [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]\delta x(t)$$

$$\text{where} \quad M_1 = 2m_1 l_1, \; M_2 = 2m_2 l_2, \; \text{and} \; Z = 4M + M_1 + M_2.$$

Figure 2: Linearized model of the dual pendulum system, adopted from [1].

which takes the standard linear form $\dot{x} = Ax+Bu$, $y = Cx+Du$ with the state vector $x = [z \; \dot{z} \; \theta_1 \; \dot{\theta_1} \; \theta_2 \; \dot{\theta_2}]'$ as visualized in Figure 1.

2 Controller Design

2.1 Controllability Analysis

We begin our analysis by evaluating the controllability of the system defined by the model in Figure 2. The controllability matrix for the system is formulated as $\kappa = [B \mid AB \mid A^2B \mid A^3B \mid A^4B \mid A^5B]$, which is performed in MATLAB (section 5). It is shown that the controllability matrix is full rank [2] and the system is controllable with the parameters provided in section 1.

To better understand the controllability of the system, we examine the rank of the controllability matrix while sweeping the length of the second pendulum l_2 with a constant mass density. The rank of the controllability matrix is examined by plotting the ratio of its smallest and largest singular values - a ratio of singular values close to 0 indicates a singular, and therefore non-full rank, controllability matrix (Figure 3).

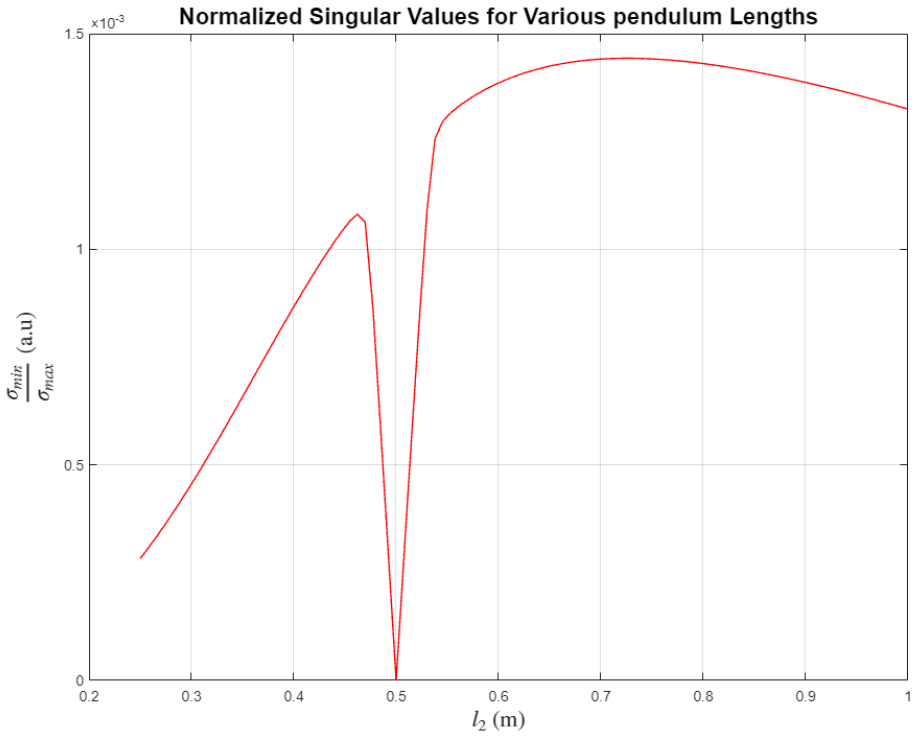


Figure 3: Ratio of the singular values of the controllability matrix for various pendulum lengths. Note that the system is not controllable for the case $l_1 = l_2$.

Note that the system becomes uncontrollable when $l_2 = l_1$, in this case $l_2 = 0.5$ m. This behavior is expected intuitively since the controller would not be able to leverage any asymmetry in the dynamics to compensate for the under-actuated configuration of the system. This study highlights an important design consideration, namely that the pendulum moments of inertia need to be unequal (and preferably substantially different) to make the system easily controllable with the current actuation scheme. We note that even if l_2 is slightly different than l_1, the controllability matrix has a nearly-zero singular value, resulting in a larger control effort required for regulation. In this exercise, this issue is avoided by choosing $l_2 = 1$ m

2.2 Optimal Controller Design with Full-State Feedback

We next design a full-state feedback Linear Quadratic Regulator (LQR) to achieve the control goals outlined above. The optimization criteria fulfilled by the controller is minimizing the following cost function:

$$J_{LQR} = \int_0^\infty ||z(t)||^2 + \rho||u(t)||^2 \, dt \tag{1}$$

where $z(t)$ is the output vector that is regulated by the controller, $u(t)$ is the control input, and ρ is a weighing parameter that allows us to balance the relative value of control effort and regulation efficacy. Optimal regulation according to equation (1) is achieved via proportional action of a matrix K on the full state, as shown in Figure 4 [2].

The first decision in the controller design is the content of the output vector $z(t)$. We choose this vector to be of the form

$$z = Gx + Hu = [z_1 \ \gamma\dot{z_1} \ \theta_1 \ \gamma\dot{\theta_1} \ \theta_2 \ \gamma\dot{\theta_2}]^T \tag{2}$$

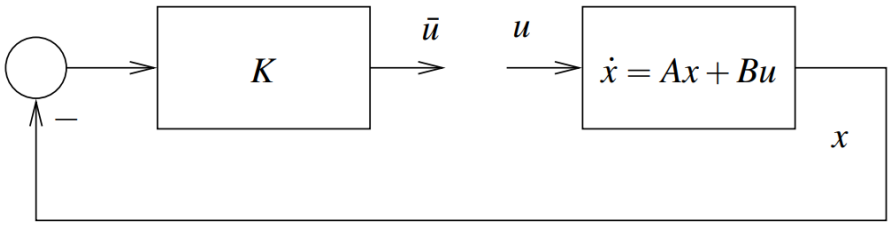


Figure 4: Block diagram of the control system, adopted from [3].

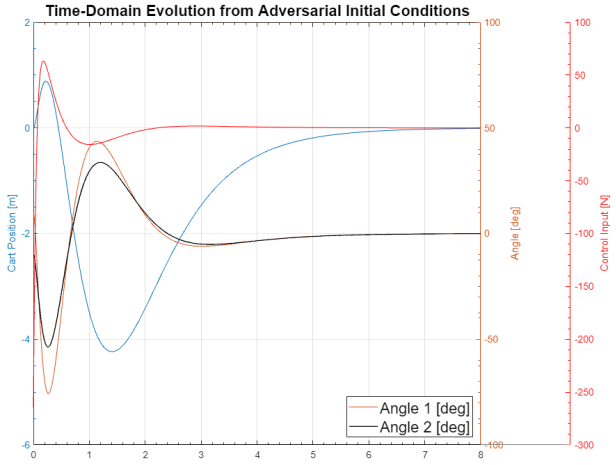
where γ is a design parameter. This choice of G and H satisfies the Kalman inequality [3] since $H = 0$, which in turn limits overshoot by providing a guarantee on the system's phase margin. Moreover, a condition for the stability of the LQR controller is detectability of the plant from the output, which is trivially fulfilled by this choice of z.

We next turn our attention to the remaining design parameters γ and ρ. As mentioned above, ρ provides a trade-off between regulation and control costs. This implies that the value of ρ should be selected to provide sufficient response time and disturbance rejection while limiting the control input magnitude to what a physical system could readily provide. The choice of γ provides a rough trade-off between maximum overshoot and response time.

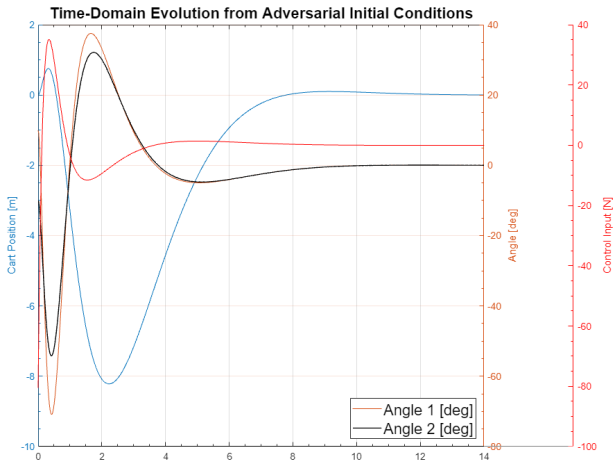
To implement this controller, the MATLAB function **lqr(sys,Q,R,N)** is used to solve the Algebraic-Riccati associated with the optimization problem and return the proportional feedback gain K. In order to recover the LQR formulation shown in equation (1), the **lqr** arguments are selected as $Q = G'G$, $R = H'H + \rho I$, and $N = G'H$. The values of the design parameters are then heuristically varied and the closed-loop performance (corresponding to the plant $A - BK$) is evaluated against two benchmark cases:

- The initial deviation of the pendulum bars is 10 degrees in opposite directions.
- Initial bar deviations are both zero, but the cart is subjected to an impulsive disturbance force of magnitude 30N and duration of 0.01 s.

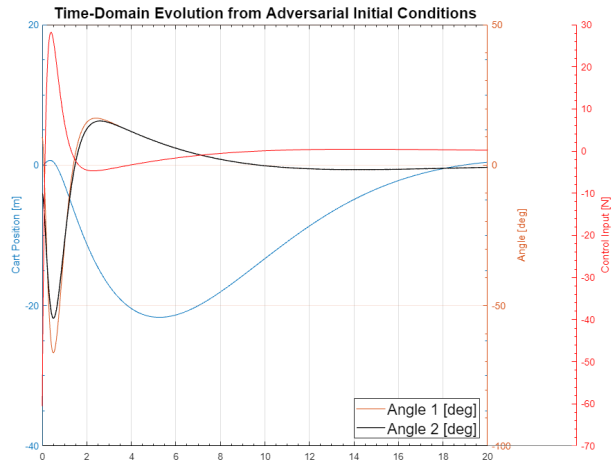
Both test cases are simulated in MATLAB. The first, unforced benchmark case is simulated using the MATLAB function **[y,tOut,x] = initial(sys,xinit)** which provides the system's unforced time-domain evolution from given initial conditions. The second case is simulated using the function **[y,tOut,x] = lsim(sys,u,t,xinit)**, which provides the system's forced time-domain evolution with the input sequence (u,t). Note that in this case, the input term Bu only contains the disturbance, since the control input is already represented in the closed-loop plant.



(a) $\rho = 0.01$

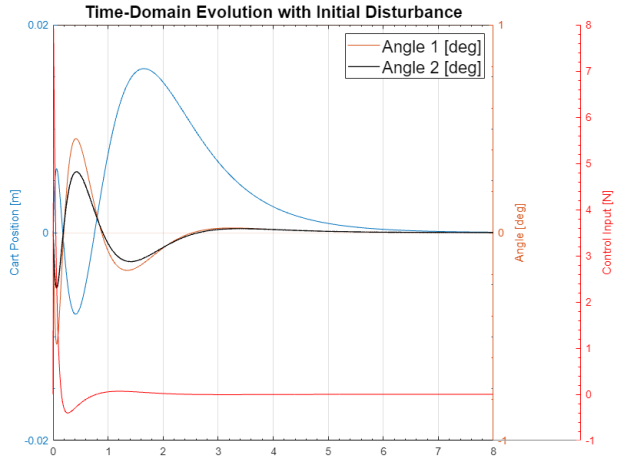


(b) $\rho = 1$

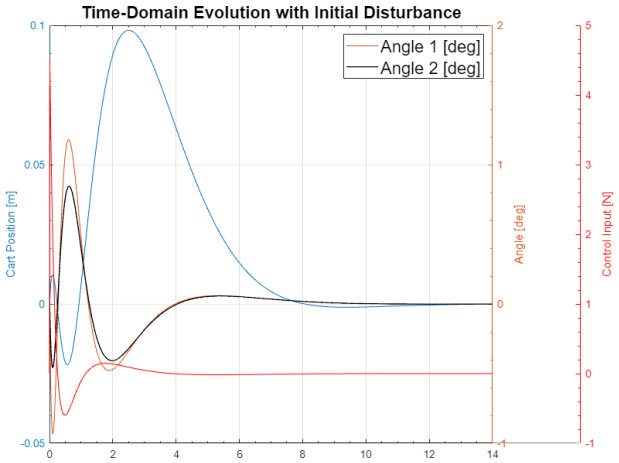


(c) $\rho = 100$

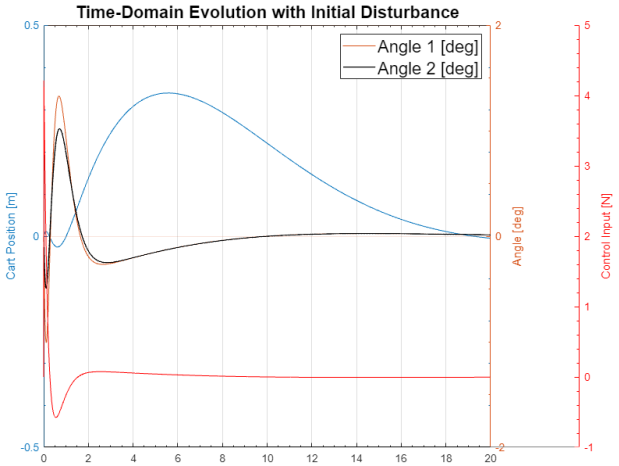
Figure 5: System response to adversarial initial conditions for various values of ρ.



(a) $\rho = 0.01$



(b) $\rho = 1$



(c) $\rho = 100$

Figure 6: System response to transient disturbance for various values of ρ.

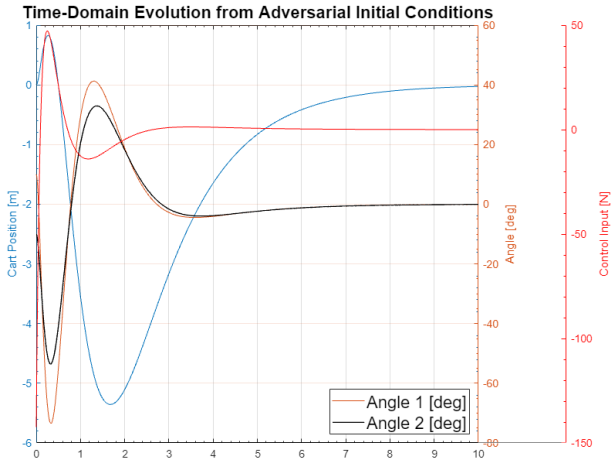
Figures 5 and 6 show the closed-loop system response to both benchmark tests for various values of the

design parameter ρ with γ fixed to a value of 1. Parametric studies like this were used to determine the design parameters that would result in satisfactory controller performance. Two specific trends can be observed from these selected studies: firstly, increasing the value of ρ prioritizes a small control cost over aggressive regulation. As a result, a controller with a larger value of ρ will result in a smaller control effort and less overshoot but admit a larger response time and poor disturbance rejection as evident in Figure 6. Using the nomenclature of frequency-domain controller design, the effect of ρ is therefore equivalent to scaling the open-loop Bode magnitude plot.

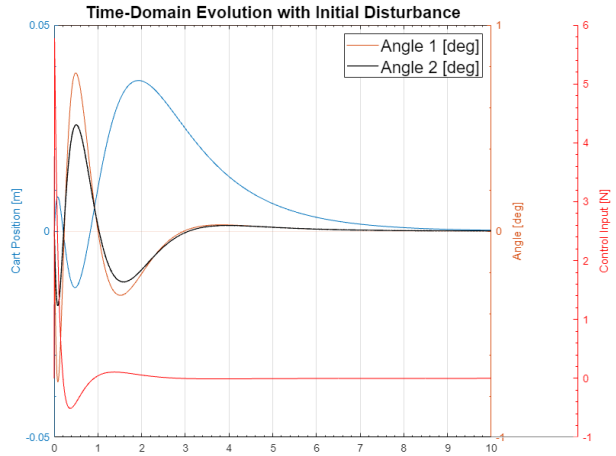
In this design task, a small value of ρ is advantageous since it improves response time and disturbance rejection. However, the resulting control signals must be small enough to be easily recreated in hardware. To verify this physical constraint, consider the maximum control effort required for the case $\rho = 0.01$, which is $u_{max} = 300$ N. A survey of commercial off-the-shelf DC motors (Polulu Corporation, Las Vegas, Nevada, USA) suggests that an output torque of $\tau = 4.5$ Nm can be readily extracted. For a rack-and-pinion design, the pinion diameter required to exert the required tangential force equals $d = \frac{2 \cdot \tau}{u_{max}} = 3$cm. Given the dimensions of the pendulum, this appears to be a reasonable first-order design. Similarly, a survey of medium-duty linear actuators suggests this is a manageable loading condition. However, to reduce hardware cost and provide a larger factor of safety while staying within the rough design goals established in section 1 we choose $\rho = 0.1$ for the final controller design.

A similar procedure was repeated to choose the second design parameter as $\gamma = 2.5$. The performance of the final controller with respect to both benchmark tests is shown in Figure 7. This discussion is qualitative due to the open-ended constraints on the system, but the procedure outlined above highlights how the controller and actuator can be changed to either provide improved closed-loop performance or reduce cost and complexity.

We note a final concern with the simulated system response: there is an appreciable angular overshoot in the first benchmark case due to the aggressive control action. In hardware implementation, this can cause the system to exit the linearized dynamics regime which, in turn, invalidates the controller analysis above. Additional hardware fine-tuning of ρ and γ may be needed to address this failure mode.



(a) System response to adversarial initial conditions.



(b) System response to transient disturbance.

Figure 7: Closed-loop performance of the final system in both benchmark tests.

2.3 Optimal Control Analysis

In addition to providing the optimal feedback gain K, the **lqr()** routine also returns the solution to the Algebraic Riccati Equation, P. The following relationship is true for the LQR performance objective:

$$min_u(\int_0^\infty x^T Q x + u^T R u \, dt \,) = x_0^T P x_0 \tag{3}$$

where x_0 is the initial condition. The implication of equation (3) is that the **lqr()** routine also returns some information about the control effort required for different initial conditions. Moreover, the Rayleigh-Ritz inequality shows that the normalized cost functional $J(x) = \frac{x^T P x}{x^T x}$ attains its maximum and minimum values when x is chosen to be the eigenvector corresponding to the largest and smallest eigenvalues of P respectively.

Using the **eig(A)** routine in MATLAB, we can find a list of all of the eigenvalues and eigenvectors associated with P. The discussion above highlights that the eigenvectors corresponding to the largest and smallest eigenvalues are directional vectors in state space that correspond to the initial conditions which

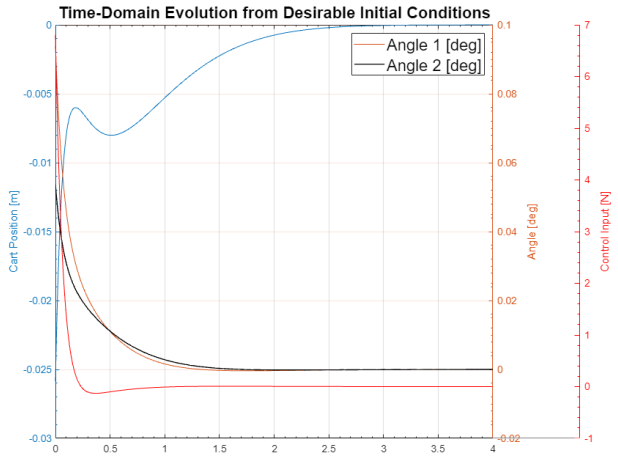
are the hardest and easiest to regulate. For the final controller implemented in section 2.2, the following eigenvalue-vector pairs are found for P:

$$\begin{aligned} \lambda_{min} &= 0.4079 \\ v_{\lambda_{min}} &= [-0.0259\ 0.3772\ 0.0938\ -0.8311\ 0.0536\ -0.3933]^T \\ \lambda_{max} &= 1.0885e04 \\ v_{\lambda_{max}} &= [0.0079\ 0.0287\ -0.5169\ -0.1330\ 0.7927\ 0.2929]^T \end{aligned} \tag{4}$$

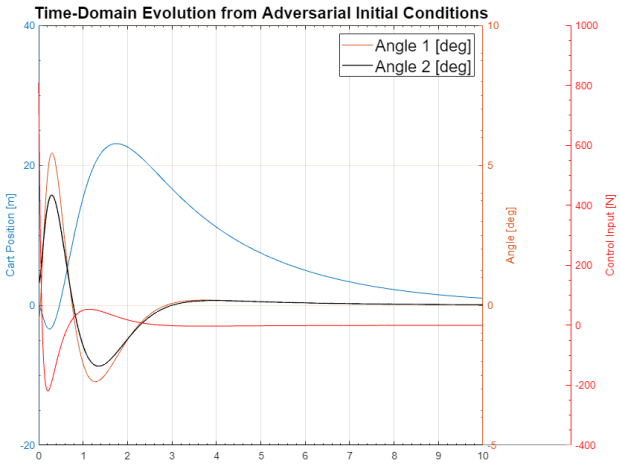
(5)

Figure 8 shows the system response for the two extreme initial conditions. We note that the most difficult initial condition to regulate is similar to the first benchmark test in 2.2, namely when the two bars point in opposite directions and therefore require different control inputs for stabilization. Moreover, the bars have adversarial angular velocities tending to pull them further apart. An inspection of the system's evolution shows that the controller first stabilizes the shorter pendulum, which can be understood by noting the larger natural frequency, and therefore the faster dynamics, of the short bar.

In contrast, the easiest initial condition to regulate is the case of the bars pointing in the same direction and having angular velocities with an opposite sense to this perturbed position. The cart position and velocity similarly aid the stabilization of the bars with a single control action, i.e. by returning the cart to the origin. The controller forces in this case are 2 orders of magnitude smaller than the adversarial case discussed above and applied over a shorter interval, highlighting the vast difference in the effort required for regulation. This discrepancy shows the importance of analyzing the required control effort for specific initial conditions that the designer may expect to regulate frequently in the physical system.



(a) System response to desirable initial conditions.



(b) System response to adversarial initial conditions.

Figure 8: Closed-loop performance of the final system for initial conditions corresponding to the eigenvectors of the P matrix. Note the relative magnitudes of the control input for the two cases.

3 Observer Design

3.1 Observability Analysis

We next consider the case in which the system's states are not all measured, which is a common occurrence with angular and linear encoders. Specifically, we check the rank of the observability matrix ($o = [C \mid AC \mid A^2C \mid A^3C \mid A^4C \mid A^5C]'$) for the following outputs using MATLAB:

- Available measurements are cart displacement, θ_1, and θ_2.
- Available measurements are cart displacement and θ_1 only.
- Available measurements are cart displacement and θ_2 only.
- Available measurements are cart displacement and $\theta_1 - \theta_2$.

The observability matrices for all cases are found to be full-rank. This outcome may be initially surprising for the cases in which only one angle is measured; however, the dynamics of z, θ_1, and θ_2 all depend on

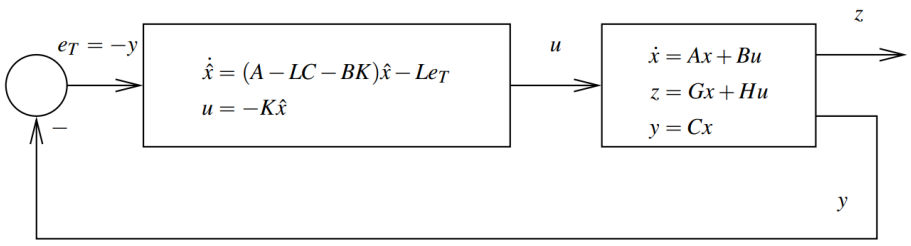


Figure 9: Block diagram of the observer-based control system, adopted from [3].

each other, meaning that knowing two of the variables provides enough information to extrapolate the third variable along with its derivative.

3.2 Optimal Observer-Based Controller Design

Finally, we design an observer-based controller for the system using z, θ_1, and θ_2 as the available measurements. Instead of assuming the full state of the system to be available via measurements, the controller designed in this section uses a dynamic estimation of the full state provided by an observer based on the available measurements. As seen in Figure 9, this observer is a linear estimator based on the correction matrix L:

$$\dot{\hat{x}} = A\hat{x} + Bu + L(\hat{y} - y) \tag{6}$$

where $\hat{x}$ and $\hat{y}$ are estimated values of the state and the output respectively. A common optimal approach to designing L is to choose a matrix that minimizes the expected value of estimation error for zero-mean Gaussian noise processes (known as the Linear-Quadratic-Gaussian (LQG) problem). The optimal matrix L is found using the MATLAB function **[est,L,P] = kalman(sys,Q,R,N)**, where Q, R, and N are weighing matrices that represent the relative magnitude of various process noises.

To maintain the system response achieved with LQR design, it is desirable to select the smallest values of weighing matrices that approximate the expected noise levels. More specifically, we choose the form $Q = I$, $R = \sigma I$ where σ is a design parameter that represents the relative value of measurement noise. A small value of σ implies small measurement noises relative to the observer uncertainty and, in turn, fast L dynamics which recovers the response seen from the full-state controller.

In this case, we start with an estimated $\sigma = 0.01$ corresponding to the angular resolution of the encoder associated with the motor specified in section 2.2. The full system can be formulated as a 12-dimensional, unforced state-space model which contains the states of the physical system x as well as the estimated states $\hat{x}$, represented below in matrix partition notation:

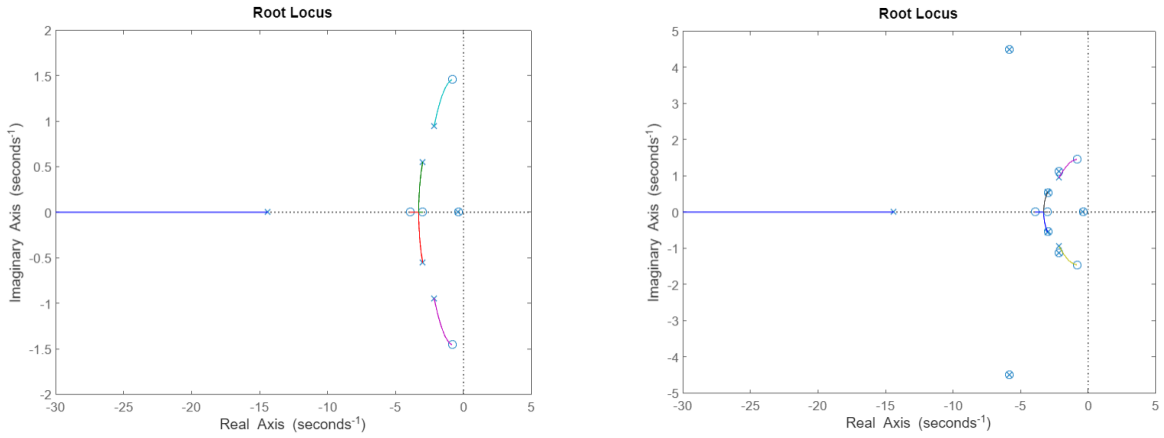
$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} A & -BK \\ LC & A - LC - BK \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} \tag{7}$$

$$y = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix}$$

Note that the partitioned state matrix captures the full system dynamics including the control action; however, any disturbance would still propagate through the system using the B matrix of the model in Figure 2.

Figures 11 and 12 show the system response to the two benchmark cases for various values of σ around this initial estimate. As expected, smaller values of σ result in a system response that more closely approximates the full-state feedback response - in this case, we select $\sigma = 0.01$. In practice, larger values of σ enhance the high-frequency response of the closed-loop system [3], so a larger σ may be chosen if the design is intended to fulfill specific high-frequency performance metrics.

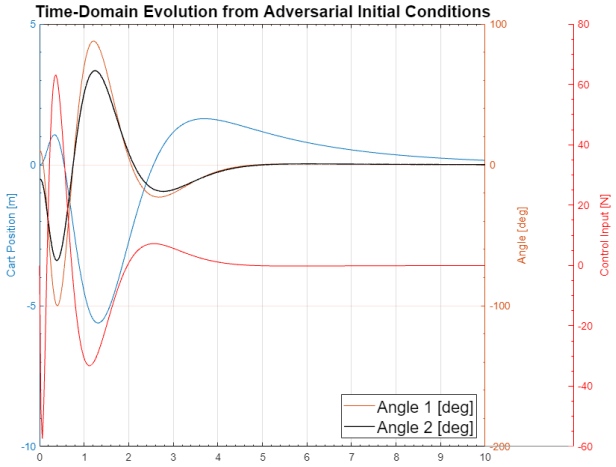
Inspecting the eigenvalues of the observer plant $(A - LC)$ shows the presence of a slow oscillatory mode close to the imaginary axis. From the eigenvalue separation principle, it is known that the set of eigenvalues of the full system is equal to the union of the sets of eigenvalues of the controller and observer plants. Subsequently, this slow mode will be present in the full system and justifies the increased response time and overshoot of Figure 11 relative to Figure 5. The poles and zeros of the full-state feedback controller and observer-based controller (Figure 10) reflect this difference.



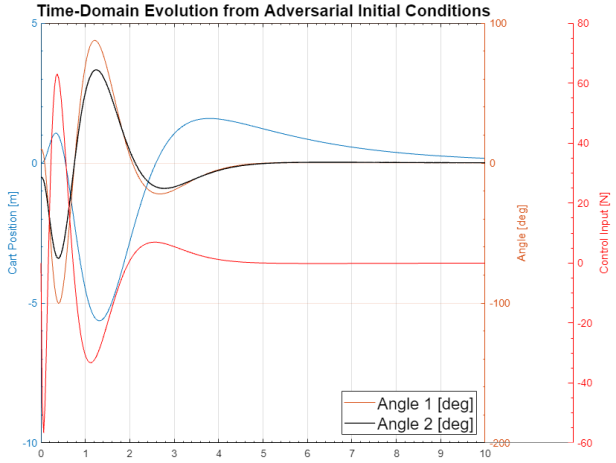
(a) Full-state feedback controller.

(b) Observer-based controller.

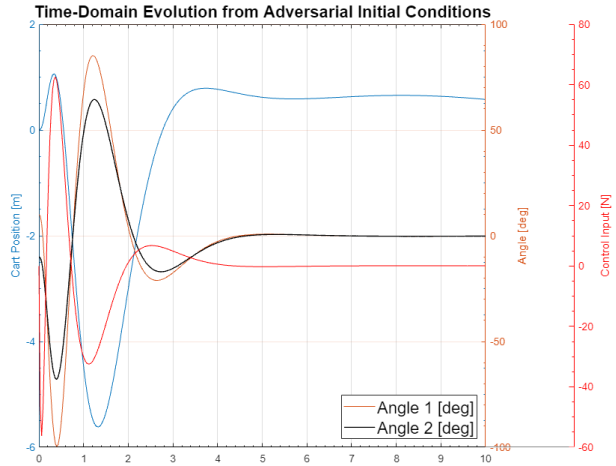
Figure 10: The poles and zeros of the full-state feedback controller and observer-based controller displayed using the **rlocus()** routine in MATLAB.



(a) $\sigma = 0.001$

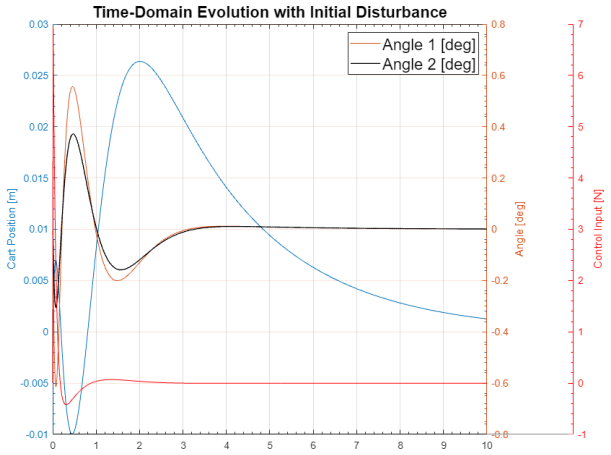


(b) $\sigma = 0.1$

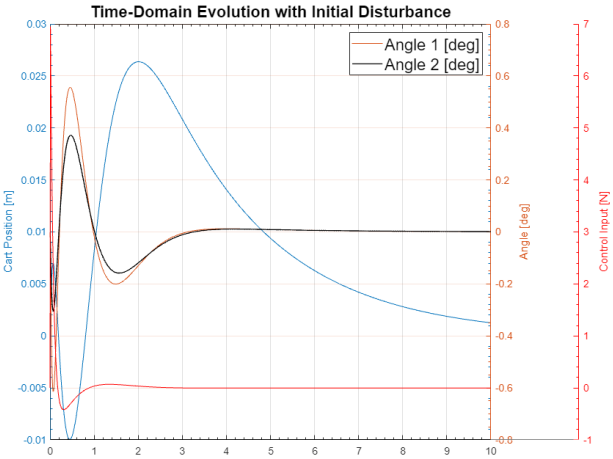


(c) $\sigma = 10$

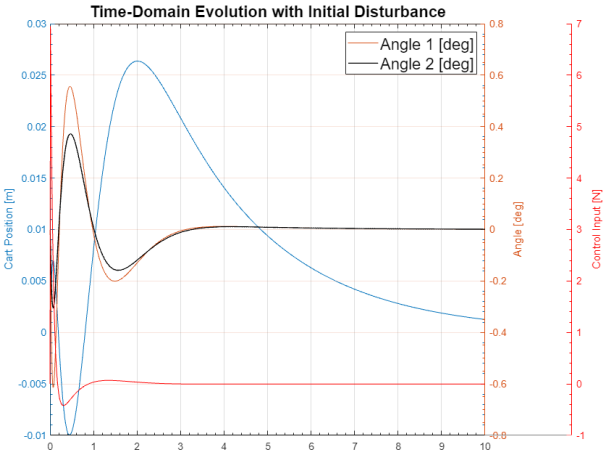
Figure 11: System response to adversarial initial conditions for various values of σ.



(a) $\sigma = 0.001$



(b) $\sigma = 0.1$



(c) $\sigma = 10$

Figure 12: System response to transient disturbance for various values of σ.

3.3 Controller Input-Output Transfer Function

One feature of interest for the control system shown in Figure 9 is the input-output transfer function of the observer-based controller which is the Laplace-domain map between the output y and the control input u. To obtain the transfer function, we start with the state-space model of the controller:

$$\left[\dot{\hat{x}}\right] = \left[A - LC - BK\right]\left[\hat{x}\right] + \left[L\right] y$$

$$u = \left[-K\right]\left[\hat{x}\right]$$

note that this model is contained in the full-system model. Once the state-space model is obtained, the individual transfer functions between the controller inputs (z, θ_1, and θ_2) and outputs (u) can be computed using the MATLAB routine **[b,a] = ss2tf(A,B,C,D)**. Because the controller is a multiple-input single-output system, the function needs to be called separately for each input variable.

```
zToU =

    -157.8 s^5 + 138 s^4 + 2.985e04 s^3 + 7.292e04 s^2 - 5.249e04 s - 9414
  ------------------------------------------------------------------------
  s^6 + 47.08 s^5 + 992.1 s^4 + 6842 s^3 + 2.234e04 s^2 + 3.835e04 s + 2.788e04
```

(a) $\frac{U}{Z}$

```
theta1ToU =

  -1.131e-13 s^5 - 1.493e-12 s^4 - 6.539e-12 s^3 + 6.915e-12 s^2 + 7.083e-11 s + 1.066e-10
  ----------------------------------------------------------------------------------------
       s^6 + 47.08 s^5 + 992.1 s^4 + 6842 s^3 + 2.234e04 s^2 + 3.835e04 s + 2.788e04
```

(b) $\frac{U}{\Theta_1}$

```
theta2ToU =

  7676 s^5 + 1.35e05 s^4 + 1.051e06 s^3 + 4.071e06 s^2 + 7.598e06 s + 5.719e06
  ----------------------------------------------------------------------------
  s^6 + 47.08 s^5 + 992.1 s^4 + 6842 s^3 + 2.234e04 s^2 + 3.835e04 s + 2.788e04
```

(c) $\frac{U}{\Theta_2}$

Figure 13: Input-output transfer functions.

Figure 13 shows the relevant transfer functions - it is separately verified that the transfer functions have negative-real poles. Moreover, all transfer functions are in 6th order and strictly proper as expected.

4 Conclusion

In summary, we first presented a full-state feedback LQR controller design for the inverted double-pendulum along with corresponding controllability and control effort analysis. We then used this development to create an observer-based LQR/LQG controller for the system under realistic measurement constraints. In addition to the hardware fine-tuning required to implement these controllers, future development of this design can include a robustness analysis of the controller with respect to plant and input uncertainty.

5 Appendix 1: MATLAB Script

```
%ME243A, Final Project - Vedad Bassari
clear; clc; close all;

%% System Definition
M = 1; %Cart mass [kg]
m1 = 0.2; %Bar mass [kg]
m2 = m1;
l1 = 0.5; %Bar length [m]
l2 = 1;
g = 9.81; %Acceleration of gravity [m/s^2]

%Compund variables
M1 = 2*m1*l1;
M2 = 2*m2*l2;
Z = 4*M + M1 + M2;

%% Controllability Analysis
%Define variables that need to be swept
l2Sweep = linspace(0.25,1,100);
m2Sweep = linspace(0.05,0.2,100);
M2Sweep = 2.*m2Sweep.*l2Sweep;
ZSweep = 4*M + M1 + M2Sweep;

sRatio = zeros(1,length(l2Sweep)); %Allocate storage for singular values

for i = 1:length(l2Sweep) %Sweep through different values of l2 and m2

    %State matrix
    A = [ 0 , 1, 0, 0, 0, 0; ...
         0, 0, (-3*M1*g)/ZSweep(i), 0, (-3*M2Sweep(i)*g)/ZSweep(i), 0; ...
         0, 0, 0, 1, 0, 0; ...
         0, 0, (3*g*(ZSweep(i)+(3*M1)))/(4*ZSweep(i)*l1), 0, ...
         (9*M2Sweep(i)*g)/(4*ZSweep(i)*l1), 0; ...
         0, 0, 0, 0, 0, 1; ...
         0, 0, (9*M1*g)/(4*ZSweep(i)*l2Sweep(i)), 0, ...
         (3*g*(ZSweep(i)+(3*M2Sweep(i))))/(4*ZSweep(i)*l2Sweep(i)), 0 ];

    B = [ 0; 4/ZSweep(i); 0; -3/(ZSweep(i)*l1);...
         0; -3/(ZSweep(i)*l2Sweep(i))]; %Input matrix

    %Formulate the controllability matrix
    A1 = A*B;
```

```
    A2 = (A^2)*B;
    A3 = (A^3)*B;
    A4 = (A^4)*B;
    A5 = (A^5)*B;
    K = [B A1 A2 A3 A4 A5];

    s = svd(K);
    sRatio(i) = min(s)/max(s);
end

r = rank(K); %Check that controllability matrix is full rank for l2 = 1 m.

FS = 16;
figure(1);
plot(l2Sweep,sRatio,'r','LineWidth',1); %Position
ylabel('$\frac{\sigma_{min}}{\sigma_{max}}$ (a.u)','Interpreter',...
    'latex','FontSize',FS);
xlabel('$l_2$ (m)','Interpreter','latex','FontSize',FS);
title('Normalized Singular Values for Various pendulum Lengths',...
    'FontSize',FS);
grid on;

%% Controller Design

    %State matrix
    A = [ 0 , 1, 0, 0, 0, 0; ...
         0, 0, (-3*M1*g)/Z, 0, (-3*M2*g)/Z, 0; ...
         0, 0, 0, 1, 0, 0; ...
         0, 0, (3*g*(Z+(3*M1)))/(4*Z*l1), 0, ...
         (9*M2*g)/(4*Z*l1), 0; ...
         0, 0, 0, 0, 0, 1; ...
         0, 0, (9*M1*g)/(4*Z*l2), 0, ...
         (3*g*(Z+(3*M2)))/(4*Z*l2), 0 ];

    B = [ 0; 4/Z; 0; -3/(Z*l1);...
         0; -3/(Z*l2)]; %Input matrix

    C = [1, 1, 1, 1, 1, 1]; %Output matrix for full-state feedback

    D = 0; %Feedforward matrix

    %Formulate state-space model
    sys = ss(A, B, C, D);
```

```
%LQR parameters
rho = 0.1;
gamma = 2.5;

%Output matrices for LQR controller
G = [1, 0, 0, 0, 0, 0;...
    0, gamma, 0, 0, 0, 0;...
    0, 0, 1, 0, 0, 0;...
    0, 0, 0, gamma, 0, 0;...
    0, 0, 0, 0, 1, 0;...
    0, 0, 0, 0, 0, gamma];

H = [0; 0; 0; 0; 0; 0];

%LQR arguments
Q = G'*G;
R = H'*H + rho;
N = G'*H;

%Call LQR function
[K,S,P] = lqr(sys,Q,R,N);
C = K;

%Formulate the closed-loop system
ACL = A - B*K; %Closed-loop plant model
sysCL = ss(ACL, B, C, D); %Closed-loop state-space model

%Simulate time evolution for benchmark #1
[y1,tOut1,x1] = initial(sysCL,[0,0,deg2rad(10),0,deg2rad(-10),0],10);
%Simulate time evolution for benchmark #2
t = 0:0.01:10;
u = zeros(1,length(t));
u(1) = 30; %Define initial disturbance [N]
[y2,tOut2,x2] = lsim(sysCL,u,t,[0,0,0,0,0,0]);

x1(:,3:6) = rad2deg(x1(:,3:6)); %Convert angles to degrees
x2(:,3:6) = rad2deg(x2(:,3:6)); %Convert angles to degrees

%Plot simulation results
figure(2);
labels = {'Cart Position [m]'; 'Angle [deg]'; 'Control Input [N]'};
plotyyy(tOut1,x1(:,1),tOut1,x1(:,3),tOut1,y1,labels);
```

```
hold on;
plot(tOut1,x1(:,5),'k','LineWidth',1);
title('Time-Domain Evolution from Adversarial Initial Conditions',...
    'FontSize',FS);
xlabel('t (s)','Interpreter','latex','FontSize',FS);
legend('Angle 1 [deg]','Angle 2 [deg]','FontSize',FS,'location',...
    'southeast');
grid on;

figure(3);
labels = {'Cart Position [m]'; 'Angle [deg]'; 'Control Input [N]'};
plotyyy(tOut2,x2(:,1),tOut2,x2(:,3),tOut2,y2,labels);
hold on;
plot(tOut2,x2(:,5),'k','LineWidth',1);
title('Time-Domain Evolution with Initial Disturbance',...
    'FontSize',FS);
xlabel('t (s)','Interpreter','latex','FontSize',FS);
legend('Angle 1 [deg]','Angle 2 [deg]','FontSize',FS,'location',...
    'northeast');
grid on;

%% Control Effort Study

[vP, dP] = eig(S);

%Simulate time evolution for the best IC
[yB,tOutB,xB] = initial(sysCL,vP(:,1),4);
    %Simulate time evolution for the worst IC
[yW,tOutW,xW] = initial(sysCL,vP(:,end),10);

%Plot simulation results
figure(3);
labels = {'Cart Position [m]'; 'Angle [deg]'; 'Control Input [N]'};
plotyyy(tOutB,xB(:,1),tOutB,xB(:,3),tOutB,yB,labels);
hold on;
plot(tOutB,xB(:,5),'k','LineWidth',1);
title('Time-Domain Evolution from Desirable Initial Conditions',...
    'FontSize',FS);
xlabel('t (s)','Interpreter','latex','FontSize',FS);
legend('Angle 1 [deg]','Angle 2 [deg]','FontSize',FS,'location',...
    'northeast');
grid on;
```

```
figure(4);
labels = {'Cart Position [m]'; 'Angle [deg]'; 'Control Input [N]'};
plotyyy(tOutW,xW(:,1),tOutW,xW(:,3),tOutW,yW,labels);
hold on;
plot(tOutW,xW(:,5),'k','LineWidth',1);
title('Time-Domain Evolution from Adversarial Initial Conditions',...
    'FontSize',FS);
xlabel('t (s)','Interpreter','latex','FontSize',FS);
legend('Angle 1 [deg]','Angle 2 [deg]','FontSize',FS,'location',...
    'northeast');
grid on;

%% Observability Analysis

%Case 1: z, theta1, theta2 are measured
%Form observability matrix
C1 = [1, 0, 0, 0, 0, 0;...
    0, 0, 1, 0, 0, 0;...
    0, 0, 0, 0, 1, 0;];
O1 = [C1; C1*A; C1*A^2; C1*A^3; C1*A^4; C1*A^5;];
rankO1 = rank(O1); %Show that observability matrix is full rank

%Case 2: z, theta1 are measured
%Form observability matrix
C2 = [1, 0, 0, 0, 0, 0;...
    0, 0, 1, 0, 0, 0;];
O2 = [C2; C2*A; C2*A^2; C2*A^3; C2*A^4; C2*A^5;];
rankO2 = rank(O2); %Show that observability matrix is not full rank

%Case 3: z, theta2 are measured
%Form observability matrix
C3 = [1, 0, 0, 0, 0, 0;...
    0, 0, 0, 0, 1, 0;];
O3 = [C3; C3*A; C3*A^2; C3*A^3; C3*A^4; C3*A^5;];
rankO3 = rank(O3); %Show that observability matrix is full rank

%Case 4: z, theta1 - theta2 are measured
%Form observability matrix
C4 = [1, 0, 0, 0, 0, 0;...
    0, 0, 1, 0, -1, 0;];
O4 = [C4; C4*A; C4*A^2; C4*A^3; C4*A^4; C4*A^5;];
rankO4 = rank(O4); %Show that observability matrix is full rank
```

```
%% Observer-Based Controller Design
%LQG parameters
sigma = 0.001;

%Output matrices for LQG observer
C = [1, 0, 0, 0, 0, 0;...
    0,0,0,0,0,0;...
    0, 0, 1, 0, 0, 0;...
    0,0,0,0,0,0;...
    0, 0, 0, 0, 1, 0;...
    0,0,0,0,0,0;];
BB = B;
sysLQG = ss(A, [B BB], C, 0);
Q = sigma.*1; %Single input
R = sigma.*eye(3); %Three outputs
N = 0; %Assume noises uncorrelated

%Call LQG function
[est,L,P] = kalman(sysLQG,eye(),sigma.*eye());

%Formulate closed-loop system
ACLObserver = [A, -B*K; L*C, A - L*C - B*K;];
BCLObserver = [B; B;]; %Account for disturbances using B
CCLObserver = [0,0,0,0,0,0,K]; %Reutrn control input
DCLObserver = 0;
sysCLObserver = ss(ACLObserver,BCLObserver,CCLObserver,DCLObserver);

%Simulate time evolution for benchmark #1
[y1,tOut1,x1] = initial(sysCLObserver,[0,0,deg2rad(10),0,...
    deg2rad(-10),0,0,0,0,0,0,0],10);
%Simulate time evolution for benchmark #2
t = 0:0.01:10;
u = zeros(1,length(t));
u(1) = 30; %Define initial disturbance [N]
[y2,tOut2,x2] = lsim(sysCLObserver,u,t,[0,0,0,0,0,0,0,0,0,0,0,0]);

x1(:,3:6) = rad2deg(x1(:,3:6)); %Convert angles to degrees
x2(:,3:6) = rad2deg(x2(:,3:6)); %Convert angles to degrees

%Plot simulation results
figure(5);
labels = {'Cart Position [m]'; 'Angle [deg]'; 'Control Input [N]'};
plotyyy(tOut1,x1(:,1),tOut1,x1(:,3),tOut1,y1,labels);
```

```
hold on;
plot(tOut1,x1(:,5),'k','LineWidth',1);
title('Time-Domain Evolution from Adversarial Initial Conditions',...
    'FontSize',FS);
xlabel('t (s)','Interpreter','latex','FontSize',FS);
legend('Angle 1 [deg]','Angle 2 [deg]','FontSize',FS,'location',...
    'southeast');
grid on;

figure(6);
labels = {'Cart Position [m]'; 'Angle [deg]'; 'Control Input [N]'};
plotyyy(tOut2,x2(:,1),tOut2,x2(:,3),tOut2,y2,labels);
hold on;
plot(tOut2,x2(:,5),'k','LineWidth',1);
title('Time-Domain Evolution with Initial Disturbance',...
    'FontSize',FS);
xlabel('t (s)','Interpreter','latex','FontSize',FS);
legend('Angle 1 [deg]','Angle 2 [deg]','FontSize',FS,'location',...
    'northeast');
grid on;

%Compute transfer function of the controller
C = [1, 0, 0, 0, 0, 0;...
    0,0,0,0,0,0;...
    0, 0, 1, 0, 0, 0;...
    0,0,0,0,0,0;...
    0, 0, 0, 0, 1, 0;...
    0,0,0,0,0,0;];
Actrl = A - L*C - B*K;
Bctrl = L;
Cctrl = -K;
Dctrl = zeros(1,6);
[b1,a1] = ss2tf(Actrl,Bctrl,Cctrl,Dctrl,1);
[b2,a2] = ss2tf(Actrl,Bctrl,Cctrl,Dctrl,2);
[b3,a3] = ss2tf(Actrl,Bctrl,Cctrl,Dctrl,3);

%Formulate transfer functions
zToU = tf(b1,a1);
theta1ToU = tf(b2,a2);
theta2ToU = tf(b3,a3);

%Verify transfer function poles
pZToU = pole(zToU);
```

```
pTheta1ToU = pole(theta1ToU);
pTheta2ToU = pole(theta2ToU); %All poles are negative-real

%Verify transfer function zeros
sZToU = zero(zToU);
sTheta1ToU = zero(theta1ToU);
sTheta2ToU = zero(theta2ToU); %All poles are negative-real

%Plot the root-locus of both controllers
figure(7);
rlocus(sysCL); %Full-state feedback
figure(8);
rlocus(sysCLObserver); %Observer-based
```

References

[1] L. C. Phillips, *Control of a dual inverted pendulum system using linear-quadratic and H-infinity methods.* Master's Thesis, Massachusetts Institute of Technology, 1994. [Online]. Available: https://dspace.mit.edu/handle/1721.1/36507

[2] K. Ogata, *Modern Control Engineering*, ser. Instrumentation and controls series. Prentice Hall, 2010. [Online]. Available: https://books.google.com/books?id=Wu5GpNAelzkC

[3] J. Hespanha, *Advanced Undergraduate Topics in Control Systems Design*, 2023. [Online]. Available: https://web.ece.ucsb.edu/ hespanha/published/allugtopics-20230402.pdf